

# Rukovanje ulazom i izlazom

Ulaz i izlaz ili I/O predstavlja komunikaciju između procesirajućeg sistema – npr. računara – i spoljnog sveta. Ulaz je signal koji taj sistem prima, a izlaz je signal koji taj sistem šalje. Signale šalju uređaji ili drugi sistemi.

I/O proces je na većini modernih računara asinhron (nezavisan) u smislu da je vreme izvršavanja operacija na centralnoj procesorskoj jedinici (CPU) i I/O uređaja nezavisno. Kada je reč o CPU-u, on je po prirodi sinhron (istovremen), odnosno, kontrolna jedinica izvršava svaku naredbu/instrukciju pod vremenom koje određuje centralni sat u procesoru. Tako se svaka operacija izvršava u tačno određeno vreme i sa tačno određenim trajanjem, jer svaka naredba u mašinskom kodu (kodu koji se jedino može izvršavati na procesoru) traje unapred definisan broj otkucaja procesorskog sata.

Kada je reč o vremenima izvršenja na I/O uređajima, ne važe ista pravila. Ulazno/izlazne instrukcije samo iniciraju operaciju, dok vreme izvršenja operacije zavisi od fizičke prirode uređaja i stanja u kojem se on nalazi. Takođe, može zavisiti i od mehaničkih operacija unutar uređaja koje su potrebne da bi se izvršile zadate instrukcije, koje ne mogu biti tako precizne kao elektronska kola unutar procesora ili drugih čipova. Ovakav problem se rešava konstrukcijom kontrolera – posebnog elektronskog kola koje će funkcionisati kao most između procesora i samog uređaja, pa tako, procesor nema direktnu komunikaciju sa I/O uređajem, već zapravo komunicira sa tim kontrolerom.

Pošto smo rekli da ulazna ili izlazna instrukcija samo započinju interakciju – svaki uređaj može biti samo u tri stanja: zauzeto, stanje čekanja i isključen. Dakle, ukoliko nije isključen, može biti ili u stanju zauzetosti prethodnom komandom ili u stanju čekanja na sledeću. Inicijalno, stanje uređaja je uvek – čekanje. Kada procesor izvrši I/O naredbu koja se odnosi na određeni uređaj, taj uređaj prelazi iz stanja čekanja u stanje zauzetosti i u tom stanju ostaje dokle god tu naredbu ne završi. Nakon što je završi, uređaj prelazi u stanje čekanja i spreman je za sledeću I/O komandu. Procesor takođe vodi računa i o zauzetosti I/O uređaja.

## I/O uređaji

Većina uređaja je ili ulaznog ili izlaznog tipa, što znači da ili mogu da šalju podatke koje korisnik prosleđuje ili inicira ili se ti podaci mogu vraćati nazad korisniku. Ali, postoji i manja grupa uređaja koji su i ulaznog i izlaznog tipa, odnosno ulazno-izlazni uređaji. Neki od ulaznih, izlaznih i ulazno-izlaznih uređaja su:

### ulazni uređaji:

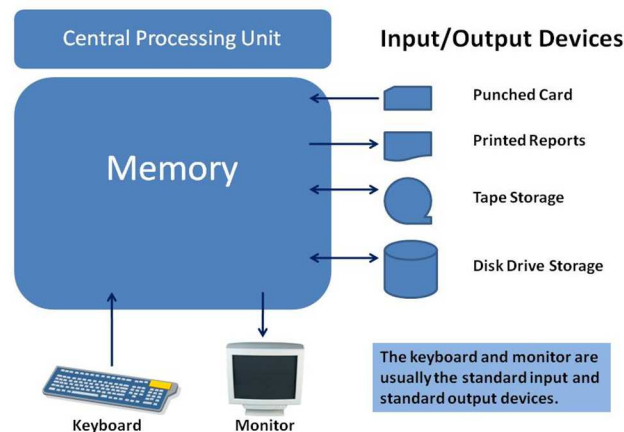
- tastatura i miš – prihvataju ulazne naredbe korisnika i prosleđuju ih računaru; ne mogu poslati signal nazad korisniku;
- mikrofoni – prima zvuk i šalje ga računaru ili drugom sistemu na obradu;
- web kamera – prima sliku preko senzora i šalje ih računaru ili drugom sistemu na obradu;

### izlazni uređaji:

- monitor – prima podatke u vidu teksta, slika itd. od računara i prikazuje ih na ekranu; osim u slučaju ekrana osetljivog na dodir, ne može primiti korisnički unos;
- projektor – prima izlaz sa računara u vidu slika i projektuje ih u vidu slika i teksta na zadatu površinu; ne može primiti ulazne instrukcije od korisnika niti ih slati;
- zvučnici – primaju podatke u vidu zvuka, ali ne mogu primiti zvuk kao ulaznu informaciju korisnika;

### ulazno-izlazni uređaji:

- DVD-RW dražv – čita diskove (ulaz) i prima podatke sa računara i upisuje ih na disk (izlaz);
- USB dražv – prima ili snima podatke sa računara (ulaz) ili te podatke, sa fleša, šalje računaru (izlaz).



Slika 8.1. Šematski prikaz nekih I/O uređaja<sup>1</sup>

#### Pitanje

U kojem od ponuđenih stanja ne može biti I/O uređaj?

- **Hibernacija**
- Čekanje
- Isključen

#### Objašnjenje:

Tačan odgovor je da I/O po definiciji ne može biti u stanju hibernacije; ako je uključen, može biti ili zauzet ili u stanju čekanja na sledeću naredbu.

<sup>1</sup> <https://press.rebus.community/programmingfundamentals/chapter/input-and-output/>

## Ulazni/izlazni koncept u Tkinteru

Kada je reč o Tkinteru, ulazni uređaji su ti kojima upravljamo kontrolama i događajima u Tkinteru. To su najčešće miš i tastatura, ali mogu biti i ekrani osetljivi na dodir. Sa druge strane, ekran je najčešće uređaj koji je zadužen za prikaz izlaznih/povratnih informacija.

Za jedan I/O primer u Tkinteru možemo uzeti scenario gde se korisniku dozvoljava da pomoću strelica na tastaturi (ulaz) pomera kvadrat na ekranu (izlaz). Ovo se može postići pomoću događaja, gde ćemo pri svakom okidanju događaja `KeyPress` proveriti koji taster je pritisnut i na osnovu toga pomeriti kvadrat na tu stranu. Pomeranje ćemo izvršiti metodom `.move()`. Takođe, ako naš kvadrat dođe bilo koje ivice – ne želimo da nastavi da ide dalje, *izvan* prozora aplikacije, već na suprotnu stranu.

Kako bismo dodali ovu funkcionalnost, pri svakom okidanju `KeyPress` događaja, moramo voditi računa trenutnoj poziciji kvadrata na platnu i uz to ga pomerati. Za to nam je potrebna metoda `.coords()`, koja se poziva nad canvas objektom i kojoj se prosleđuje objekat čije koordinate želimo. Povratna vrednost poziva `.coords()` je lista sa četiri elementa – koordinate: `x1,y1,x2,y2`. Pored ove funkcionalnosti, metodom `coords()` možemo i zadati nove koordinate objektu na canvasu. U tom slučaju prosleđujemo ime objekta (njegov id) i n-torku ili listu sa onoliko koordinata kao elemenata koliko je potrebno da se zadati objekat opiše.

U našem primeru, ako je reč o levoj i desnoj ivici prozora aplikacije, vodićemo računa da `x1` vrednost ne pređe ili 0 (leva ivica) ili vrednost promenljive `WIDTH` koja predstavlja maksimalnu širinu prozora aplikacije. Slično je i kada je reč o gornjim i donjim ivicama – za njih ćemo najviše obratiti pažnju na to da vrednost koordinate `y1` ne pređe ili 0 (gornja ivica) ili vrednost promenljive `HEIGHT` (donja ivica).

Takođe, pri pokretanju programa želimo da postavimo ekran na sredinu ekrana. Ovo možemo učiniti tako što za `x` i `y` vrednosti uzmemo polovine `WIDTH` (`x1 = WIDTH / 2`) i `HEIGHT` (`y1 = HEIGHT / 2`) vrednosti koje predstavljaju dužinu i širinu prozora aplikacije.

### Primer korišćenja strelica na tastaturi za pomeranje objekta u Tkinteru

```
import tkinter as tk

root = tk.Tk()

WIDTH = 500
HEIGHT = 500

canvas = tk.Canvas(root, width = WIDTH, height = HEIGHT)
canvas.pack()

x1 = WIDTH / 2
y1 = HEIGHT / 2

# Center on x axis, at the bottom of y axis
text = canvas.create_text(x1, y1 * 2 - 20,
                          text = "Use arrow keys to move")
image = canvas.create_rectangle(x1, y1, x1 + 15, y1 + 15)

def move(event):
```

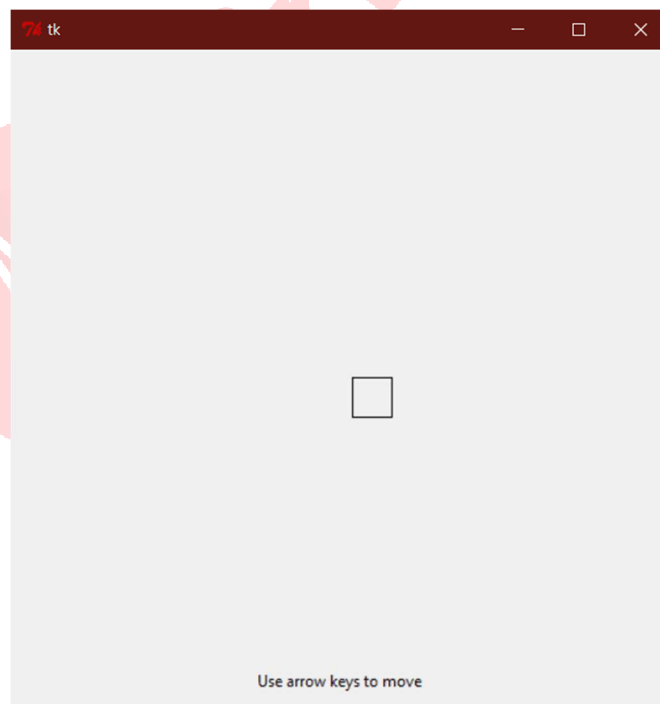
```

if event.keysym == "Left":
    current_coords = canvas.coords(image)
    if current_coords[0] <= 0:
        canvas.move(image, WIDTH, 0)
    else:
        canvas.move(image, -10, 0)
elif event.keysym == "Right":
    current_coords = canvas.coords(image)
    if current_coords[0] >= WIDTH:
        canvas.move(image, -WIDTH, 0)
    else:
        canvas.move(image, 10, 0)
elif event.keysym == "Up":
    current_coords = canvas.coords(image)
    if current_coords[1] <= 0:
        canvas.move(image, 0, HEIGHT)
    else:
        canvas.move(image, 0, -10)

elif event.keysym == "Down":
    current_coords = canvas.coords(image)
    if current_coords[1] >= HEIGHT:
        canvas.move(image, 0, -HEIGHT)
    else:
        canvas.move(image, 0, 10)

root.bind("<KeyPress>", move)
root.mainloop()

```



*Animacija 8.1. Primer pokretanja vidžeta na osnovu korisničkih instrukcija*

Izmeniti primer tako da se umesto kvadrata komandama pomera pravougaonik. Dopuniti primer da ako se klikne na taster Enter(događaj „Return“) program pauzira na 4 sekunde i potom zatvori prozor(quit metoda nad postojećim canvas objektom).

## Rezime

- Ulaz i izlaz ili I/O predstavlja komunikaciju između procesirajućeg sistema – npr. računara – i spoljnog sveta. Ulaz je signal koji taj sistem prima, a izlaz je signal koji taj sistem šalje. Signale šalju uređaji ili drugi sistemi.
- Ulazno/izlazne instrukcije samo iniciraju operaciju, dok vreme izvršenja operacije zavisi od fizičke prirode uređaja i stanja u kojem se on nalazi. Takođe može zavisiti i od mehaničkih operacija unutar uređaja koje su potrebne da bi se izvršile zadate instrukcije, koje ne mogu biti tako precizne kao elektronska kola unutar procesora ili drugih čipova.
- Svaki uređaj može biti samo u tri stanja: zauzeto, stanje čekanja i isključen. Dakle, ukoliko nije isključen, može biti ili u stanju zauzetosti prethodnom komandom ili u stanju čekanja na sledeću. Inicijalno, stanje uređaja je uvek – čekanje. Kada procesor izvrši I/O naredbu koja se odnosi na određeni uređaj, taj uređaj prelazi iz stanja čekanja u stanje zauzetosti i u tom stanju ostaje dokle god tu naredbu ne završi. Nakon što je završi, uređaj prelazi u stanje čekanja i spreman je za sledeću I/O komandu.
- Kada je reč o Tkinteru, ulazni uređaji su ti kojima upravljamo kontrolama i događajima u Tkinteru; to su najčešće miš i tastatura, ali mogu biti i ekrani osetljivi na dodir.
- Sa druge strane, ekran je najčešće uređaj koji je zadužen samo za prikaz izlaznih/povratnih informacija.

